Universitat
Autònoma
de Barcelona

5725: Codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Elena Cuevas Cobo
i dirigit per
Joaquim Borges Ayats
Bellaterra, 13 de Juny de 2014

El signant, Joaquim Borges i Ayats, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha estat realitzada sota la seva direcció per Elena Cuevas Cobo

Bellaterra, 13 de Juny de 2014

Signat: Joaquim Borges i Ayats

Als meus companys,
que han esdevingut la meva família en aquests 6 anys

Agraïments

Especialment al meu tutor, Quim Borges, per les seves constants correccions i el seu entusiasme perquè aquest projecte quedés perfecte. Al Josep Rifà, per preocupar-se també del curs del projecte i ajudar-me a definir un enfocament. I finalment a tota la gent del meu entorn, per la paciència que han tingut aquests mesos.

Índex

Índex de figures

Índex de taules

1 Introducció

1.1 Codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$

Aquest projecte se sustenta en la Teoria de Codis i més concretament parlarà de codis propelineals invariants per translacions, que també anomenarem codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.[1]

L'increment de les comunicacions durant el segle XX va motivar un fort desenvolupament de la Teoria de Codis i sobretot dels codis lineals. Per la seva banda, l'any 1989, en Josep Rifà introdueix els codis propelineals, arran de la publicació de l'article [5]. Els codis propelineals són una generalització dels codis lineals que tenen una estructura algebraica compatible amb la mètrica d'Hamming, però que no són necessàriament lineals.

L'any 1994 surt publicat un nou article que parla sobre els codis $\mathbb{Z}4$-lineals [4]. No serà però fins més tard que es descobreixi que aquests són un cas particular dels codis propelineals.

A partir d'aquí comença a créixer l'interès per codis (lineals o no) que tenen unes característiques molt determinades. Sobretot s'estudien els codis $\mathbb{Z}_2$-lineals (codis lineals) i $\mathbb{Z}_4$-lineals i, més en general, els codis $\mathbb{Z}_2\mathbb{Z}_4$-lineals [1]. Els dos darrers són particularment interessants ja que, igual que passava amb els codis lineals, disposen d'eines com matriu generadora, matriu de control, etc.

Més endavant, al 1997, en Jaume Pujol publicarà la seva tesi (juntament amb l'article [6]), que dirigirà Josep Rifà i que parlarà de forma més extensa de codis propelineals que són invariants per translacions.

A partir de llavors es comença a sentir curiositat per aquests codis, els codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$, que estan un nivell per sobre dels $\mathbb{Z}_2\mathbb{Z}_4$-lineals quant a generalització. Al principi es pensava que aquests podrien tenir tots estructura del tipus $\mathbb{Z}_2\mathbb{Z}_4$ i, de fet, prenent només una dimensió de $\mathcal{Q}_8$ així és. No obstant això, el professor A. del Rio descobreix un exemple de codi amb una part $\mathcal{Q}_8$ que no podia tenir estructura de $\mathbb{Z}_2\mathbb{Z}_4$. [3]

Fins ara no s'ha volgut aprofundir en l'estudi dels codis que inclouen el conjunt dels

[1] Els codis propelineals invariants per translacions són isomorfs als codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$

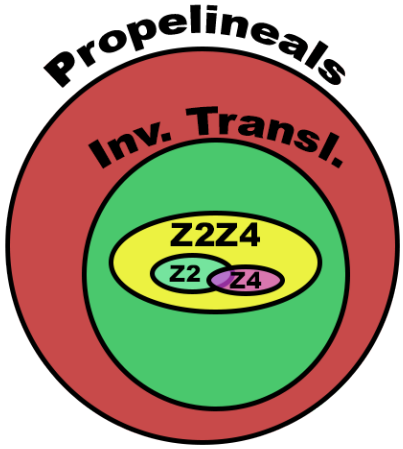

Figura 1: Classificació dels codis propelineals

quaternions ja que, donat que el grup no és abelià, no permeten parlar de dualitat, i això els fa poc aplicables en casos pràctics. La seva estructura, però, fa pensar en codis de Hadamard. Actualment la recerca en aquest camp se centra en buscar codis interessants d'aquests tipus. Aquest serà doncs el nostre punt de partida.

1.2 Objectius

L'objectiu principal del projecte és el d'estudiar els codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ i en especial els codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. Primerament s'analitzarà en profunditat la teoria sobre aquests per tal de, posteriorment, implementar funcions que ens defineixin les característiques d'un codi introduït. Concretament voldrem saber, a través d'aquestes funcions:

- Si un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ és Hadamard o no.
- El seu *shape*
- El seu *rank*
- El seu *kernel*

Posteriorment, es considerarà la possibilitat d'implementar alguna de les construccions recursives conegudes d'aquest tipus de codis.

A més a més, es redactarà una memòria sobre el projecte que inclourà tant els resultats teòrics necessaris com l'explicació de les implementacions; amb la previsió que algú que hi estigui interessat pugui, posteriorment seguir amb la recerca. Amb aquesta finalitat es tractarà de fer una documentació autocontinguda i suficient per a qualsevol que vulgui seguir-la.

1.3 Contingut de la memòria

La memòria es composa de 7 capítols:

- **Capítol 1: Introducció.** En aquest capítol inicial es fa una aproximació al tema d'estudi, els codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. També s'informa sobre l'estat de l'art i els objectius del projecte.

- **Capítol 2: Fonaments Teòrics.** En aquest capítol es donen les bases matemàtiques i de Teoria de Codis que s'han hagut d'assolir per a la realització de les funcions que s'establien als objectius del projecte. El lector trobarà informació que s'ha publicat sobre els codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ en format matemàtic.

- **Capítol 3: Planificació del Projecte.** En aquest capítol es descriu el procés que se seguirà per al correcte desenvolupament del projecte en els límits de temps establerts. El lector podrà observar les diverses tasques a desenvolupar en un format estàndard, establert per a la planificació de projectes.

- **Capítol 4: Entorn de Desenvolupament.** En aquest capítol s'expliquen les característiques de l'entorn sobre el qual s'ha desenvolupat el projecte. Es dóna informació sobre la màquina utilitzada i l'entorn de programació. A més, es descriuen de manera breu els estàndards de programació que s'han utilitzat i les característiques específiques del llenguatge triat. El lector també podrà trobar una breu comparativa entre les tres possibilitats existents d'entorn de desenvolupament més importants: GAP, Magma i Sage.

- **Capítol 5: Desenvolupament del Projecte.** En aquest capítol es descriuen els algorismes usats per a programar les funcions més importants del desenvolupament del projecte. El lector podrà tenir una visió general del paquet de funcions així com entrar en detall sobre l'algorísmica i els criteris de programació utilitzats.

- **Capítol 6: Resultats.** En aquest capítol es mostren els resultats obtinguts amb el desenvolupament de la llibreria.

- **Capítol 7: Conclusions.** En aquest capítol s'exposen les conclusions obtingudes durant el desenvolupament del projecte, així com possibles millores i ampliacions a desenvolupar en el futur.

2 Fonaments teòrics

2.1 Preliminars

2.1.1 Aplicacions de Gray

Siguin $\mathbb{Z}_2$ i $\mathbb{Z}_4$ el cos binari i l'anell d'enters mòdul 4 respectivament. Sigui $\mathbb{Z}_2^n$ el conjunt de tots els vectors binaris de longitud n i $\mathbb{Z}_4^n$ el conjunt de totes les n-tuples sobre l'anell $\mathbb{Z}_4$.

Definició 1. El *Pes d'Hamming* d'un vector $v \in \mathbb{Z}_2^n$ denota el número de coordenades diferents de zero de v. Escrivim $wt(v)$.

Definició 2. Definim la *Distància d'Hamming* entre dos vectors $u, v \in \mathbb{Z}_2^n$ com

$$d(u, v) = wt(u + v).$$

Definició 3. Qualsevol subconjunt no buit de $\mathbb{Z}_2^n$ s'anomena un *codi binari* i, similarment, un subespai lineal de $\mathbb{Z}_2^n$ s'anomena un *codi lineal binari* o *codi* $\mathbb{Z}_2$*-lineal*. De la mateixa manera, un subconjunt no buit de $\mathbb{Z}_4^n$ és un *codi quaternari* i un subgrup de $\mathbb{Z}_4^n$ s'anomena un *codi lineal quaternari*.

Observació 1. Els codis $\mathbb{Z}_4$ poden ser vistos com codis binaris sota una Aplicació de Gray definida com

$$\begin{array}{rccc} \varphi: & \mathbb{Z}_4 & \to & \mathbb{Z}_2 \times \mathbb{Z}_2 \\ & 0 & \mapsto & (0,0) \\ & 1 & \mapsto & (0,1) \\ & 2 & \mapsto & (1,1) \\ & 3 & \mapsto & (1,0) \end{array}$$

Aquesta aplicació pot ser estesa, aplicada coordenada per coordenada, mitjançant una bijecció

$$\tilde{\varphi}: \ \mathbb{Z}_4^n \ \to \ \mathbb{Z}_2^{2n},$$

de manera que si $\mathcal{C}$ és un codi quaternari lineal de longitud n aleshores el codi $\mathcal{C} = \tilde{\varphi}(C)$ és un codi $\mathbb{Z}_4$-lineal de longitud binària $2n$.

Sigui $\mathcal{Q}_8$ el grup de quaternions de vuit elements. Una representació de $\mathcal{Q}_8$ és

$$\mathcal{Q}_8 = \langle a, b : a^4 = a^2b^2 = 1, bab^{-1} = a^{-1}\rangle = \left\{1, a, a^2, a^3, b, ab, a^2b, a^3b\right\}.$$

Un *codi quaterniònic* $\mathcal{C}$ és un subgrup no buit de $\mathcal{Q}_8^n$.

Observació 2. Un codi quaterniònic pot ser representat com un codi binari sota la següent Aplicació de Gray:[2]

$$\begin{array}{rccc} \phi: & \mathcal{Q}_8 & \to & \mathbb{Z}_2^4 \\ & 1 & \mapsto & (0,0,0,0) \\ & a & \mapsto & (0,1,0,1) \\ & a^2 & \mapsto & (1,1,1,1) \\ & a^3 & \mapsto & (1,0,1,0) \\ & b & \mapsto & (0,1,1,0) \\ & ab & \mapsto & (1,1,0,0) \\ & a^2b & \mapsto & (1,0,0,1) \\ & a^3b & \mapsto & (0,0,1,1) \end{array}$$

Denotem per $\tilde{\phi}$ l'extensió component a component de ϕ de $\mathcal{Q}_8^n$ a $\mathbb{Z}_2^{4n}$. Si $\mathcal{C}$ és un codi quaterniònic, aleshores $C = \tilde{\phi}(\mathcal{C})$ és un codi $\mathcal{Q}_8$ de longitud binaria $4n$.

Definició 4. Donats $k_1, k_2, k_3 \in \mathbb{N} \cup \{0\}$, definim l'*aplicació de Gray generalitzada*

$$\begin{array}{rccc} \Phi: & \mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3} & \to & \mathbb{Z}_2^{k_1+2k_2+4k_3} \\ & (x, y, z) & \mapsto & (x, \tilde{\varphi}(y), \tilde{\phi}(z)) \end{array}$$

d'aquesta manera tenim que un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ és un codi binari C de la forma $C = \Phi(\mathcal{C})$, on $\mathcal{C}$ és un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.

[2] Aquí ens permetem fer un abús de llenguatge, ja no es pot dir que aquesta aplicació sigui una Aplicació de Gray, donat que no conserva la distància 1 entre els seus punts.

Observació 3. $\mathcal{C}$ és abelià si i només si C és un codi $\mathbb{Z}_2\mathbb{Z}_4$-lineal, és a dir, si $k_3 = 0$.

D'ara en endavant utilitzarem la notació additiva per a $\mathbb{Z}_2$ i $\mathbb{Z}_4$ i la multiplicativa per a $\mathcal{Q}_8$. Per tant, $\mathbf{e} = (0, \overset{k_1+k_2}{\ldots}, 0, 1, \overset{k_3}{\ldots}, 1)$ denotarà l'element neutre de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$. Notem, a més, que cadascun dels grups $\mathbb{Z}_2$, $\mathbb{Z}_4$ i Q_8 tenen un únic element d'ordre 2. Aleshores existeix un únic element $\mathbf{u}$ de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ que té l'element d'ordre 2 corresponent a cada coordenada. A més a més, aquest element queda determinat perquè el vector $\Phi(\mathbf{u})$ és el vector que té un 1 a totes les coordenades.

Definició 5. Sigui $h \in \mathbb{Z}$, $w = (x, y, z) \in \mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$. Aleshores

$$w^h = (hx, hy, z^h).$$

Definició 6. L'*ordre* de $w \in \mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ és l'enter positiu més petit h tal que $w^h = e$.

2.1.2 Codis isomorfs i equivalents

Definició 7. Sigui S_n el grup simètric de permutacions sobre n elements. Sigui $\pi \in S_n$ una permutació i $v = (v_1, ..., v_n) \in \mathbb{Z}_2^n$. Definim

$$\pi(v) = (v_{\pi^{-1}(1)}, ..., v_{\pi^{-1}(n)}).$$

Definició 8. Diem que dos codis C_1, C_2 binaris de longitud n són *isomorfs* si existeix una permutació $\pi \in S_n$ tal que

$$C_2 = \{\pi(x) : x \in C_1\}.$$

Diem que són *equivalents* si existeix un vector $y \in \mathbb{Z}_2^n$ i una permutació $\pi \in S_n$ tal que

$$C_2 = \{y + \pi(x) : x \in C_1\}.$$

Observació 4. Dos codis binaris lineals són equivalents si i només si són isomorfs.

2.1.3 Codis propelineals

Definició 9. Diem que un codi binari C de longitud n és *propelineal* si per a qualsevol paraula codi $x \in C$ existeix $\pi_x \in S_n$ que satisfaci, per a tot $v \in \mathbb{Z}_2^n$ i $x, y \in C$ les següents propietats:

1. $x + \pi_x(y) \in C$
2. $\pi_x(\pi_y(v)) = \pi_z(v)$, on $z = x + \pi_x(y)$.

Observació 5. Sigui C un codi propelineal. Per a tot $x \in C$, sigui $\pi_x \in S_n$ la permutació que satisfà les condicions anteriors. Per a tot $x \in C,\ y \in \mathbb{Z}_2^n$ tenim $xy = x + \pi_x(y)$. Això indueix una estructura de grup a C, que no és abeliana en general.

Lema 1. *Sigui C un codi propelineal de longitud n. Aleshores*

$$d(u, v) = d(xu, xv),$$

per a tot $x \in C$ i $u, v \in \mathbb{Z}_2^n$.

És a dir, la multiplicació per l'esquerra en un codi propelineal és compatible Hamming, és a dir, que

$$d(xz, x) = wt(z),$$

per a tot $x \in C$ i $z \in \mathbb{Z}_2^n$.

Definició 10. Diem que un codi propelineal C de longitud n és *invariant per translacions* si

$$d(x, y) = d(xu, yu),$$

per a tot $x, y \in C,\ u \in Z_2^n$.

Proposició 1. *[6] Un codi binari és propelineal invariant per translacions si i només si és un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.*

2.1.4 Rank i kernel per a codis binaris

Definició 11. El *rank* d'un codi binari C és la dimensió de l'espai vectorial binari generat per les seves paraules codi. El denotem per $r(C)$.

Definició 12. El *kernel* d'un codi binari C de longitud n és

$$K(C) = \{z \in \mathbb{Z}_2^n : C + z = C\}.$$

Denotem per $k(C)$ la dimensió de $K(C)$.

Observació 6. Si dos codis binaris tenen diferents ranks o dimensió del kernel aleshores no poden ser equivalents.

Lema 2. *Si C és un codi binari no lineal aleshores*

$$r(C) \geq k(C) + 2.$$

Si, a més C és un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ aleshores

$$r(C) \geq k(C) + 3.$$

2.1.5 Matrius de Hadamard

Definició 13. Una *Matriu de Hadamard* d'ordre n és una matriu $H = (h_{ij})$, $i, j \in \{1, .., n\}$ i $h_{ij} \in \{1, -1\}$ per a tot $i, j \in \{1, .., n\}$ tal que $HH^T = nId$.

Definició 14. Diem que dues matrius de Hadamard són *equivalents* si podem obtenir una a través de permutacions de files o columnes de l'altra i multiplicant files i/o columnes per -1.

2.2 Propietats dels codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$

En aquesta secció enunciarem alguna de les propietats de grup dels codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. Seguirem l'article [2] i usarem la següent notació:

- $x^y = y^{-1}xy$ *conjugat* de $x \in \mathcal{C}$ per $y \in \mathcal{C}$
- $(x, y) = x^{-1}y^{-1}xy$, *commutador* de $x, y \in \mathcal{C}$
- $\mathcal{C}' = \langle (x, y) : \ x, y \in \mathcal{C} \rangle$ *subgrup commutador* de $\mathcal{C}$
- $Z(\mathcal{C}) = \{z \in \mathcal{C} : \ zx = xz \ \forall x \in C\}$ el *centre* de $\mathcal{C}$
- $T(\mathcal{C}) = \{z \in \mathcal{C} : \ z^2 = e\}$

Observació 7. Notem que

$$\mathcal{C}' \subseteq T(\mathcal{C}) \subseteq Z(\mathcal{C}),$$

és a dir que $\mathcal{C}'$ i $T(\mathcal{C})$ són ambdós subgrups pertanyents al centre de $\mathcal{C}$, el que implica que

1. $(x, y) = (y, x)$
2. $(xy, z) = (x, z)(y, z)$ per a tot $x, y, z \in \mathcal{C}$

Definició 15. Diem que $\mathcal{C}$ és de *tipus* (σ, δ, ρ) si:

- $| \ T(\mathcal{C}) \ | = 2^\sigma$
- $[Z(\mathcal{C}) : T(\mathcal{C})] = 2^\delta$
- $[\mathcal{C} : Z(\mathcal{C})] = 2^\rho$

Observació 8. Suposem $\mathcal{C}$ del tipus (σ, δ, ρ). Aleshores $T(\mathcal{C}) \cong \mathbb{Z}_2^\sigma$.

Observació 9. Tot element $c \in \mathcal{C}$ pot ser escrit de manera única com

$$c = \prod_{i=1}^{\sigma} x_i^{\alpha_i} \prod_{i=1}^{\delta} y_j^{\beta_j} \prod_{i=1}^{\rho} z_k^{\gamma_k},$$

on $\alpha_i, \beta_j, \gamma_k \in \{0, 1\}$. Notem que els elements z_k no commuten entre ells però interpretem $\prod_{i=1}^{\rho} z_k^{\gamma_k}$ com $z_1^{\gamma_1} z_2^{\gamma_2} \cdots z_\rho^{\gamma_\rho}$.

A partir d'ara escriurem

$$\mathcal{C} = \langle x_1, ..., x_\sigma; y_1, ..., y_\delta; z_1, ..., z_\rho \rangle.$$

Lema 3. *Siguin $a, b \in \mathcal{C} \setminus T(\mathcal{C})$ aleshores:*

1. *Si $(a, b) = \boldsymbol{e}$ i $a^2 = b^2$ aleshores $ab \in T(\mathcal{C})$.*
2. *a^2, b^2 coincideixen en cada coordenada no trivial de (a, b). En particular*

$$wt\left(\phi\left((a, b)\right)\right) \leq wt\left(\phi\left(\left(a^2\right)\right)\right).$$

3. *Si $\mathcal{C}$ és de tipus (σ, δ, ρ) aleshores $\sigma \geq \delta + \min\{1, \rho\}$.*

Definició 16. El *swapper de $x, y \in \mathcal{G}$* és

$$[x, y] = \phi^{-1}\left(\phi(x) + \phi(y) + \phi(xy)\right).$$

Definició 17. Definim el **swapper de $\mathcal{C}$** com el conjunt

$$S(\mathcal{C}) = \{[x, y] : \ x, y \in \mathcal{C}\}.$$

Notem que per a obtenir el swapper a $\mathcal{G}$ n'hi ha prou amb calcular els swappers a $\mathbb{Z}_2, \mathbb{Z}_4$ i $\mathcal{Q}_8$. És trivial veure que $[x, y] = \mathbf{e} \ \ \forall x, y \in \mathbb{Z}_2$. Les següents taules descriuen els valors de tots els swappers a $\mathbb{Z}_4$ i $\mathcal{Q}_8$, respectivament:

	$0, 2$	$1, 3$
$0, 2$	0	0
$1, 3$	0	2

	$1, a^2$	a, a^3	b, a^2b	ab, a^3b
$1, a^2$	1	1	1	1
a, a^3	1	a^2	a^2	1
b, a^2b	1	1	a^2	a^2
ab, a^3b	1	a^2	1	a^2

Taula 1: Swappers a $\mathbb{Z}_4$ i $\mathcal{Q}_8$

En particular $[x, y] \in T(\mathcal{G})$, el que implica

$$\phi\left([x, y]xy\right) = \phi\left([x, y]\right) + \phi\left(xy\right) = \phi(x) + \phi(y).$$

Lema 4. *Siguin $x, y, z, t \in \mathcal{G}$ i sigui $C = \phi(\mathcal{C})$ un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. Aleshores es compleixen les següents propietats:*

(a) Si $z^2 = \boldsymbol{e}$ aleshores $[zx, y] = [x, zy] = [x, y]$ i $[z, x] = [x, z] = \boldsymbol{e}$.

(b) $[x, x - 1] = [x, x]$.

(c) $[x, y][y, x] = (x, y)$.

(d) $[x, x] = x^2$.

(e) $[x, yz] = [x, y][x, z]$ i $[xy, z] = [x, z][y, z]$.

(f) Si $\Phi(x) \in K(C)$ aleshores $[x, y] \in \mathcal{C}$ $\forall y \in \mathcal{C}$.

Demostració. Es pot comprovar fàcilment fent servir la igualtat

$$[x, y] = ([x_1, y_1], ..., [x_l, y_l])$$

i les taules dels swappers per a $\mathbb{Z}_4$ i $\mathcal{Q}_8$. □

El següent lema és conseqüència directa de la definició de swapper i de les propietats enunciades al lema anterior

Lema 5. *Sigui $C = \phi(\mathcal{C})$ un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ de tipus (σ, δ, ρ). Aleshores $\Phi\left(T(\mathcal{C})\right) \subseteq K(C)$ i per tant $\sigma \leq k(C)$.*

Lema 6. *Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ amb $l = k_1 + k_2 + k_3$ tals que $C = \Phi(\mathcal{C})$ és de tipus (σ, δ, ρ). Sigui $\mathcal{D} = \langle \mathcal{C} \cup S(\mathcal{C}) \rangle$, el grup generat per $\mathcal{C}$ i els swappers dels elements de $\mathcal{C}$. Aleshores*

1. *$\Phi(\mathcal{D})$ és l'espai vectorial de C.*
2. *$r(C) = \sigma + \delta + \rho + h$ on $h \leq min\left\{\binom{\delta+\rho}{2}, l - \sigma\right\}$.*

Proposició 2. *Sigui $\mathcal{C}$ el codi quaterniònic $\mathcal{C} = \langle (a,a),(ab,b)\rangle \leq \mathcal{Q}_8^2$. Sigui $C = \Phi(\mathcal{C})$ el codi $\mathcal{Q}_8$ corresponent. Aleshores $\mathcal{C}$ no és un codi $\mathbb{Z}_2\mathbb{Z}_4$-lineal.*

2.3 Codis de Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$

En aquesta secció ens centrarem ja en els codis de Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. El resultat principal d'aquest apartat serà el teorema 2 que ens proposa una classificació dels codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ en termes de la seva estructura.

Definició 18. Un codi binari C de longitud n s'anomena *codi 1-perfecte estès* si

$$\mid C \mid = \frac{2^{n-1}}{n}$$

i la seva distància mínima és 4.

Teorema 1. *Sigui $C = \Phi(\mathcal{C})$, on $\mathcal{C}$ és un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ amb $k_3 > 0$ i $n = k_1 + 2k_2 + 4k_3$. Aleshores:*

1. *Si C és un codi 1-perfecte aleshores $n = 7$ i $(k_1,k_2,k_3) = (3,0,1)$.*
2. *Si C és un codi estès 1-perfecte aleshores o bé $n = 8$ i $(k_1,k_2,k_3) \in \{(4,0,1),(0,2,1),(0,0,2)\}$ o bé $n = 4$ i $(k_1,k2,k_3) = (0,0,1)$.*

Lema 7. *Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ tal que $\Phi(\mathcal{C})$ és un codi Hadamard. Siguin $a,b,c \in \mathcal{C} \setminus T(\mathcal{C})$. Aleshores:*

1. *O bé $(a,b) \in \langle a^2 \rangle$ o bé $a^2 = \boldsymbol{u}$.*
2. *Si $a^2 = b^2 = c^2 \neq \boldsymbol{u}$ aleshores $\mid \langle [a,c],[b,c],T(\mathcal{C})\rangle / T(\mathcal{C}) \mid \leq 4$.*
3. *Si $a^2 = b^2 = (a,b) \neq c^2$ aleshores $\mid \langle [a,c],[b,c],T(\mathcal{C})\rangle / T(\mathcal{C}) \mid \leq 2$.*

Corol·lari 1. *Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ tal que $C = \Phi(\mathcal{C})$ és un codi Hadamard i $x_1,\ldots,x_\sigma; y_1,\ldots,y_\delta; z_1,\ldots,z_\rho$ és un conjunt de generadors de $\mathcal{C}$. Aleshores:*

1. *Per cada $t \in T(\mathcal{C})$ amb $t \neq \boldsymbol{u}$ tenim que el cardinal de*

$$\left\{z_i^2 = t \mid i = 1, \ldots, \rho\right\}$$

és com a molt 2.

2. *Si $(z_i, z_j) \neq \boldsymbol{e}$ aleshores o bé $z_i^2 = \boldsymbol{u}$ o bé $(z_i, z_j) = z_i^2$.*

3. *Si $(z_i, z_j) = \boldsymbol{e}$ aleshores $z_i^2 \neq z_j^2$.*

Observació 10. El corol·lari que acabem de veure implica que si $\mathcal{C}$ és un subgrup de $\mathcal{G}$ tal que $\Phi(\mathcal{C})$ és un codi de Hadamard i $t \neq \mathbf{u}$ aleshores un grup generador de $\mathcal{C}$ té dos $z_i's$ tals que $z_i^2 = t$.

Definició 19. Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ tal que $C = \Phi(\mathcal{C})$ és un codi Hadamard i $x_1, ..., x_\sigma; y_1, ..., y_\delta; z_1, ..., z_\rho$ és un conjunt de generadors de $\mathcal{C}$. Diem que aquest grup generador està *normalitzat* si $z_i^2 = \mathbf{u}$ per com a molt dues $i = 1, ..., \rho$ i si $z_i^2 = z_j^2 = \mathbf{u}$ amb $i \neq j$ aleshores $(z_i, z_j) = \mathbf{u}$.

Lema 8. *Cada codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ té un conjunt normalitzat de generadors.*

Lema 9. *Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ tal que $C = \Phi(\mathcal{C})$ és un codi Hadamard i $x_1, ..., x_\sigma; y_1, ..., y_\delta; z_1, ..., z_\rho$ és un conjunt de generadors de $\mathcal{C}$.*

Denotem per ϵ el nombre de parelles diferents de $z_i's$ amb el mateix quadrat. Reordenem els $z_i's$ de manera que aquells que tinguin el mateix quadrat siguin correlatius. Les afirmacions següents són certes:

1. *$\epsilon \leq 2$*

2. *Si $\epsilon = 2$ aleshores $\delta = 0$ i $\rho = 4$. En el cas que z_1^2, z_3^2 i $\boldsymbol{u}$ siguin diferents dos a dos tenim $(z_i, z_j) = \boldsymbol{e}$ i $z_i^2 z_j^2 = \boldsymbol{u}$ per a $i \in \{1, 2\}, j \in \{3, 4\}$*

3. *Si $z_i^2 = \boldsymbol{u}$ amb $i \leq 2\epsilon$ aleshores $\delta = 0$.*

4. *Sigui* $V = \{y_1, \ldots, y_\delta, z_1, z_3, \ldots, z_{2\epsilon-1}, z_{2\epsilon+1}, z_{2\epsilon+2}, \ldots, z_\rho\}$, $W = \{w^2 : w \in V\}$ *i* $U = \{u \in V : u^2 \neq \boldsymbol{u}\}$. *Aleshores* $|\langle W\rangle| \geq 2^{\delta+\rho-\epsilon-1}$ *i per tant* $\sigma \geq \delta + \rho - \epsilon - 1$. *Si, a més,* $\boldsymbol{u} \notin \langle U\rangle$ *aleshores* $|\langle W\rangle| = 2^{\delta+\rho-\epsilon}$ *i per tant* $\sigma \geq \delta + \rho - \epsilon$.

5. *(Límit superior per a* $r(C)$*) Si* $\mathcal{D} = \langle \mathcal{C} \cap S(\mathcal{C})\rangle$ *aleshores* $\mathcal{D}$ *és del tipus* $(\sigma + h, \delta, \rho)$ *i* $r(\mathcal{C}) \leq \sigma + \delta + \rho + h$ *amb*

$$h \leq \begin{cases} \epsilon + \binom{\delta+\rho-\epsilon}{2}, & si\ \epsilon \leq 1 \\ 3, & si\ \epsilon = 2 \end{cases}$$

Teorema 2. *Sigui* $\mathcal{C}$ *un subgrup de* $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ *tal que* $C = \Phi(\mathcal{C})$ *és un codi Hadamard de longitud* $n = 2^m$ *amb rang* $r = r(C)$ *i dimensió del kernel* $k = k(C)$. *Aleshores* $\mathcal{C}$ *té un conjunt generador normalitzat* $x_1, \ldots, x_\sigma; y_1, \ldots, y_\delta; z_1, \ldots, z_\rho$ *on* $m = \sigma + \delta + \rho - 1$, *satisfent una de les condicions següents:*

1. $\rho = 0$*: Aleshores* $\mathcal{C} \cong \mathbb{Z}_2^{\sigma-\delta} \times \mathbb{Z}_4^\delta$ *i* C *és un codi* $\mathbb{Z}_2\mathbb{Z}_4$ *(que podria ser* $\mathbb{Z}_4$ *lineal o no), de longitud* $n = 2^m$ *i* $m = \sigma + \delta - 1$.

   (a) *Si* C *és* $\mathbb{Z}_4$ *lineal aleshores o bé* $\delta \in \{1, 2\}$ *i* C *és lineal, o bé* $\delta \geq 3$, $k = \sigma + 1$; $r = \sigma + \delta + \binom{\delta-1}{2}$.

   (b) *Si* C *no és* $\mathbb{Z}_4$ *lineal aleshores* $\sigma > \delta$ *i o bé* $\delta \in \{0, 1\}$ *i* C *és lineal o bé* $\delta \geq 2$, $k = \sigma$ *i* $r = \sigma + \delta + \binom{\delta}{2}$.

2. $\delta = 0$, $z_1^2 = z_2^2 = (z_1, z_2) = \boldsymbol{u}$, $(z_i, z_j) = z_j^2$ *i* $(z_j, z_k) = \boldsymbol{e}$ *per a cada* $i \in \{1, 2\}$ *i* $3 \leq j, k \leq \rho$. *Aleshores* $\mathcal{C} \cong \mathbb{Z}_2^{\sigma-\rho+1} \times \left(\mathbb{Z}_4^{\rho-2} \rtimes \mathcal{Q}_8\right)$, $k \geq \sigma \geq \rho - 1$ *i* $r \leq \sigma + \rho + 1 + \binom{p-1}{2}$.

3. $\delta = 0$, $z_1^2 = \boldsymbol{u} \notin \langle z_2^2, ..., z_\rho^2\rangle \cong \mathbb{Z}_2^{\rho-1}$, $(z_1, z_i) = z_i^2$ *i* $(z_i, z_j) = \boldsymbol{e}$ *per a cada* $i \neq j$ *a* $\{2, ..., \rho\}$. *Aleshores* $\mathcal{C} \cong \mathbb{Z}_2^{\sigma-\rho} \times \left(\mathbb{Z}_4^{\rho-1} \rtimes \mathbb{Z}_4\right)$, $k \geq \sigma \geq \rho$ *i* $r \leq \sigma + \rho + \binom{\rho}{2}$.

4. $\rho = 2$, $\delta \leq 1$ *i* $z_1^2 = z_2^2 = (z_1, z_2) \neq \boldsymbol{u}$. *Aleshores* $\mathcal{C} \cong \mathbb{Z}_2^{\sigma-\delta-1} \times Z_4^\delta \times \mathcal{Q}_8$, $k \geq \sigma \geq \delta + 1$ *i* $r \leq \sigma + \delta + \rho + 1 \leq \sigma + 4$.

5. *$\delta = 0$, $\rho = 4$, $z_1^2 = z_2^2 = (z_1, z_2) = \boldsymbol{u} \neq z_3^2 = z_4^2 = (z_3, z_4)$ i $(z_i, z_j) \in \langle z_j^2 \rangle$ per a cada $i \in \{1, 2\}$ i $j \in \{3, 4\}$. Aleshores $\mathcal{C} \cong Z_2^{\sigma-2} \times (\mathcal{Q}_8 \rtimes \mathcal{Q}_8)$ i $k \geq \sigma \geq 2$; $r \leq \sigma + 7$.*

Definició 20. Si $\mathcal{C}$ és un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ tal que $C = \Phi(\mathcal{C})$ és un codi Hadamard amb un conjunt normalitzat de vectors generadors satisfent la condició i ($i \in \{1, \ldots, 5\}$) del 2 aleshores diem que $\mathcal{C}$ té *shape* i.

Corol·lari 2. *Sigui $\mathcal{C}$ un subgrup de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ de longitud 2^m i tipus (σ, δ, ρ) tal que $C = \Phi(\mathcal{C})$ és un codi Hadamard. Sigui $k = k(C)$ i $r = r(C)$. Aleshores es compleix que:*

1. *$\lceil \frac{m}{2} \rceil \leq \sigma \leq k \leq m + 1 \leq r \leq m + 1 + \binom{\delta+\rho}{2}$ i $\delta + \rho = m + 1 - \sigma \leq \lfloor \frac{m+2}{2} \rfloor$, a excepció del codi amb paràmetres $m = 5$, $\sigma = 2$, $\delta = 0$, $\rho = 4$.*

2. *Tindrem que*

$$r \leq \begin{cases} m + 1 + \binom{\frac{m+1}{2}}{2}, & \text{si } m \text{ és senar} \\ m + 2 + \binom{\frac{m}{2}}{2}, & \text{si } m \text{ és parell} \end{cases}$$

I, més concretament:

$$r - (m+1) \leq \begin{cases} \binom{\frac{m-1}{2}}{2}, & \text{si } m \text{ és senar i } \mathcal{C} \text{ té shape } 1 \\ 1 + \binom{\frac{m-1}{2}}{2}, & \text{si } m \text{ és senar i } \mathcal{C} \text{ té shape } 2 \\ \binom{\frac{m+1}{2}}{2}, & \text{si } m \text{ és senar i } \mathcal{C} \text{ té shape } 3 \\ \binom{\frac{m}{2}}{2}, & \text{si } m \text{ és parell i } \mathcal{C} \text{ té shape } 1 \text{ o } 3 \\ 1 + \binom{\frac{m}{2}}{2}, & \text{si } m \text{ és parell i } \mathcal{C} \text{ té shape } 2 \\ 1, & \text{si } \mathcal{C} \text{ té shape } 4 \\ 3, & \text{si } \mathcal{C} \text{ té shape } 5 \end{cases}$$

3 Planificació del projecte

3.1 Planificació temporal del treball

El projecte estava pensat per a portar-se a terme durant 6 mesos. Tot i la compaginació amb altres tasques com estudis o feina aquest temps finalment no s'ha vist incrementat.

La realització del projecte s'ha dividit en tres parts ben diferenciades:

- El desenvolupament teòric del projecte (120h): entendre bé les característiques dels codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. Entendre els resultats ja publicats.
- La implementació del projecte (90h): programar funcions que ajudin a entendre i a usar els resultats que s'utilitzin durant el desenvolupament teòric.
- La redacció de documentació del projecte (120h): redactar els resultats i les conclusions extretes de l'estudi, i fer-ho amb la perspectiva que algú pugui aprofitar-ne després els resultats i seguir amb l'estudi.

La idea era que aquestes tres parts es complementessin i s'executessin alhora durant la duració del projecte. El desenvolupament teòric, però, es va iniciar una mica abans i la redacció de la documentació també s'ha estès una mica en el temps un cop s'havien donat per finalitzades les altres dues parts.

A aquestes tres parts que podríem considerar les troncals del projecte caldria afegir-hi dos blocs més. Un bloc inicial que és la introducció al projecte, a tot el tema de la teoria de la codificació i en especial als codis que tracta el treball. També ens havíem proposat realitzar algunes construccions dels codis estudiats si disposàvem de temps extra. Malauradament, finalment aquest temps extra no ha existit.

La planificació es pot veure de forma detallada al Diagrama de Gantt annex a aquesta memòria.

3.2 Recursos

Els recursos utilitzats durant la realització d'aquest projecte han estat alguns de collita pròpia i d'altres subministrats pel Departament d'Enginyeria de la Informació i les Comunicacions (dEIC). Bàsicament s'han usat tres tipus de recursos.

3.2.1 Recursos documentals

Els recursos documentals utilitzats en aquest projecte han estat articles referents a Teoria de Codis i, en particular, aquells que feien referència a codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$. A la introducció de la memòria es fa un recorregut a través d'aquests articles, recollits també dins la bibliografia del projecte.

Pel que fa a la part d'implementació s'ha usat també tota la documentació online disponible per al programari Sage.[3]

3.2.2 Recursos de programari

Sage és un projecte gratuït de codi obert i hi ha dues possibles maneres de treure'n profit.

- Online: és la forma més senzilla d'usar Sage. Et permet accedir a la computació en Sage com un servei web usant només un navegador. Qualsevol còpia local de Sage pot ser configurada per oferir aquest servei.

- Local: Sage està també disponible per descarregar, això permet muntar-nos el nostre propi servidor Sage, que estarà disponible tan online com offline.

En el nostre cas utilitzarem la segona opció, tant per comoditat com per potència. Per accedir a Sage hem utilitzat el navegador Mozilla Firefox, que és un dels recomanats per al rendiment òptim de l'aplicació.

Tot i que Sage ofereix un entorn gràfic que permet desenvolupar funcions in-situ i desar-les, he decidit usar-lo en combinació amb Sublime Text, un editor de text que pot usar-se per escriure codi. Sublime Text ha estat usat amb una llicència d'avaluació gratuïta.

[3] Abans anomenat SAGE, *System for Algebra and Geometry Experimentation.*

Per al desenvolupament de documentació s'ha usat un processador de textos LaTeX anomenat Texmaker. Aquest programa és gratuït i multiplataforma, i integra moltes eines útils per a l'escriptura LaTeX.

3.2.3 Recursos de maquinari

Per al desenvolupament del projecte he fet servir el meu ordinador personal, que disposa de les següents característiques:

Processador	2,66GHz Intel Core 2 Duo
Memòria RAM	4GB 1067Mhz DDR3
Disc Dur	SSD 60 GB OWC Mercury Electra 3G
Gràfics	NVIDIA GeForce 9400M 256 MB
Sistema Operatiu	OS X 10.9.2 (Mavericks)

Taula 2: Descripció del maquinari

4 Entorn de desenvolupament

4.1 Programari matemàtic disponible

Al tractar-se d'un projecte sobre teoria de codis, treballarem sobre anells, cossos i matemàtica discreta. S'havia de buscar per tant, un software preparat per a la programació matemàtica. El programari que s'ha estudiat abans de determinar quin fariem servir ha estat:

- **GAP:** GAP és un sistema per a l'Àlgebra computacional discreta, especialment pensat per a implementar recursos de Teoria de Grups. També disposa d'alguns paquets orientats a la Teoria de Codi que podrien ser-nos d'utilitat. És gratuït i alhora és un projecte col·laboratiu que permet ésser ampliat. Està, però, una mica obsolet.

- **MAGMA:** Magma és també un sistema orientat a l'Àlgebra computacional. Aquest ha estat, però, més utilitzat al llarg dels anys i disposa de major documentació, suport i flexibilitat. Magma és programari privat i de pagament.

4.2 Entorn utilitzat: Sage

Finalment ens vàrem decantar per l'entorn de treball *Software for Algebra and Geometry Experimentation* (Sage). Sage és un projecte de codi obert que proporciona un recull d'eines que comprenen moltes branques de les matemàtiques, com per exemple àlgebra, combinatòria, càlcul, mètodes numèrics, teoria de nombres i teoria de grups.

La primera versió de Sage va néixer el 24 de Febrer del 2005 com un projecte de codi obert sota els termes de la *GNU General Public License*. Va néixer amb la idea de proveir els usuaris d'una "alternativa de codi obert a d'altres entorns ja existents com Magma, Maple, Mathematica o Matlab"[7]. El líder del projecte, William Stein, és matemàtic a la Universitat de Washington.

Sage utilitza, de base, el llenguatge de programació Python, però permet alhora im-

portar arxius escrits per a d'altres entorns, com per exemple Magma. Va ser aquest tret característic el que ens va fer decantar-nos per Sage, ja que ens oferia més possibilitats per aprofitar codi ja implementat.

5 Desenvolupament del Projecte

Al llarg d'aquest apartat explicarem en detall les funcions realitzades durant el desenvolupament del projecte. Es donarà una petita descripció, detalls sobre els seus paràmetres d'entrada i sortida i l'algorisme que segueix la funció.

5.1 Representació dels codis

Per a representar una paraula codi de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$ ho fem amb una array de $k_1 + k_2 + k_3$ posicions, on cada posició conté un element de $\mathbb{Z}_2$ o $\mathbb{Z}_4$ o $\mathcal{Q}_8$. Si apliquem un *Gray Map* sobre aquesta paraula codi, la representarem mitjançant una array binària de $k_1 + 2k_2 + 4k_3$ posicions, que podran ser 0 o 1.

De la mateixa manera, representarem un codi mitjançant un conjunt de paraules codi, és a dir mitjançant una array d'arrays.

5.2 Aplicació de Gray

En aquesta secció descriurem les diferents funcions construides i usades per a calcular l'aplicació de Gray de diferents elements de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.

5.2.1 z4_gray_map

Funció que implementa l'aplicació de Gray sobre un element de $\mathbb{Z}_4$.

Paràmetres d'Entrada

- v: element de $\mathbb{Z}_4$.

Paràmetre de Sortida

- w: element inicial transformat en format binari després d'aplicar el *Gray Map* a v o -1 si v no responia al format esperat.

Algorisme

Algorithm 1 z4_gray_map

```
1: function Z4_GRAY_MAP(v)
2:     if v == 0 then
3:         w ← [0, 0]
4:     else
5:         if v == 1 then
6:             w ← [0, 1]
7:         else
8:             if v == 2 then
9:                 w ← [1, 1]
10:            else
11:                if v == 3 then
12:                    w ← [1, 0]
13:                else
14:                    w ← −1
       return w
```

5.2.2 q8_gray_map

Funció que implementa l'aplicació de Gray sobre un element de $\mathcal{Q}_8$.

Paràmetres d'Entrada

- v: element de $\mathcal{Q}_8$.

Paràmetre de Sortida

- w: element inicial transformat en format binari després d'aplicar el *Gray Map* a v o -1 si v no responia al format esperat.

Algorisme

Algorithm 2 q8_gray_map

```
1: function Q8_GRAY_MAP(v)
2:     if v == 1 then
3:         w ← [0, 0, 0, 0]
4:     else
5:         if v == a then
6:             w ← [0, 1, 0, 1]
7:         else
8:             if v == aa then
9:                 w ← [1, 1, 1, 1]
10:            else
11:                if v == aaa then
12:                    w ← [1, 0, 1, 0]
13:                else
14:                    if v == ab then
15:                        w ← [1, 1, 0, 0]
16:                    else
17:                        if v == aab then
18:                            w ← [1, 0, 0, 1]
19:                        else
20:                            if v == aaab then
21:                                w ← [0, 0, 1, 1]
22:                            else
23:                                if v == b then
24:                                    w ← [0, 1, 1, 0]
25:                                else
26:                                    w ← −1
       return w
```

5.2.3 z2z4q8_gray_map

Funció que implementa l'aplicació de Gray sobre un element de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.

Paràmetres d'Entrada

- v: element de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.
- k_1: mida de l'espai $\mathbb{Z}_2$.

- k_2: mida de l'espai $\mathbb{Z}_4$.
- k_3: mida de l'espai $\mathcal{Q}_8$.

Paràmetre de Sortida

- w: element inicial transformat en format binari després d'aplicar el *Gray Map* a v o -1 si v no responia al format esperat.

Algorisme

Algorithm 3 z2z4q8_gray_map

function Z2Z4Q8_GRAY_MAP(v,$k1$,$k2$,$k3$)
  $w \leftarrow []$
  for $i = 0;\ i \leq k1;\ i++$ **do**
    $w.append(v[i])$
  for $i = 0;\ i \leq k2;\ i++$ **do**
    $w.append(z4\_gray\_map(v[k1 + i]))$
  for $i = 0;\ i \leq k3;\ i++$ **do**
    $w.append(q8z4\_gray\_map(v[k1 + k2 + i]))$
  if $exists(-1, w)$ **then return** -1
  return w

5.3 Codis Hadamard

En aquesta secció descriurem una funció creada per a determinar si un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ és Hadamard o no.

5.3.1 is_hadamard

Determina si un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ és un codi Hadamard.

Paràmetres d'entrada

- *matrix*: matriu de la qual es vol determinar si compleix els requisits per a ser una matriu de Hadamard.

Paràmetre de sortida

- La funció retorna *true* o $false$ segons si es compleixen o no les condicions.

Algorisme

Versió 1

Algorithm 4 is_hadamard (v1)

```
function IS_HADAMARD(matrix)
    if matrix == [] then
        raise("Empty Matrix")
    if is_square(matrix) then
        if dim(matrix) ≤ 2 then
            for row ∈ matrix do
                for element ∈ row do
                    if element != −1 AND element != 1 then return false
            return (matrix * matrix^t == dim(matrix) * identity(dim(matrix)))
        else
            if dim(matrix)%4 == 0 then
                for row ∈ matrix do
                    for element ∈ row do
                        if element != −1 AND element != 1 then return false
                return (matrix * matrix^t == dim(matrix) * identity(dim(matrix)))
            else
                return false
    else
        return false
```

Versió 2

Algorithm 5 is_hadamard (v2)

```
function IS_HADAMARD(matrix)
    if matrix == [] then
        raise("Empty Matrix")
    if is_square(matrix) then
        if dim(matrix) ≤ 2 OR dim(matrix)%4 == 0 then
            for row ∈ matrix do
                for element ∈ row do
                    if element != −1 AND element != 1 then return false
            return (matrix * matrix^t == dim(matrix) * identity(dim(matrix)))
        else
            return false
    else
        return false
```

5.4 Rank

En aquesta secció descriurem el procés per a construir una funció que determina el rang d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ donat.

5.4.1 can_be_build

Funció que determina si una paraula codi es pot construir a partir d'un conjunt de paraules codi determinat. És a dir, determina si és combinació lineal d'altres paraules codi.

Paràmetres d'entrada

- *word*: paraula codi que volem construir.
- *codeSet*: conjunt de paraules codi a partir de les quals volem construir *word*.

Paràmetre de sortida

- La funció retorna *true* o *false* segons si es pot construir o no *word* a partir de *codeSet*.

Algorisme

Algorithm 6 can_be_build

```
function CAN_BE_BUILD(word,codeSet)
    if solve(word = codeSet * x) then return true
    else
        return false
```

5.4.2 build_minimal_code_set

Funció que construeix un conjunt de paraules codi mínim a partir d'un altre. És a dir, construeix un codi que no tingui paraules codi que siguin combinació lineal d'altres.

Paràmetres d'entrada

- *code*: conjunt de paraules codi que volem minimitzar.

Paràmetre de sortida

- *minimalSet*: conjunt mínim obtingut a partir de *code*.

Algorisme

Algorithm 7 build_minimal_code_set

```
function BUILD_MINIMAL_CODE_SET(code)
    minimalSet ← code[0]
    for word ∈ code do
        if !can_be_build(word, minimalSet) then
            minimalSet.append(word)
    return minimalSet
```

5.4.3 code_rank

Funció que retorna el rang d'un codi donat.

Paràmetres d'entrada

- *code*: codi del qual volem calcular el rang.

Paràmetre de sortida

- *rank*: rang de *code*.

Algorisme

Algorithm 8 code_rank

```
function CODE_RANK(code)
    rank ← 0
    minimalCodeSet ← build_minimal_code_set(code)
    rank ← rank(minimalCodeSet)
    return rank
```

5.5 Kernel

En aquesta secció descriurem el procés per a construir una funció que determina el *kernel* d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ donat.

5.5.1 code_kernel

Funció que retorna el *kernel* d'un codi donat.

Paràmetres d'entrada

- *code*: codi del qual volem calcular el rang.

Paràmetre de sortida

- *kernel*: conjunt *kernel* del codi *code*.

Algorisme

Algorithm 9 code_kernel

```
function CODE_KERNEL(code)
    kernel ← []
    for v ∈ code do
        for w ∈ code do
            if !exists(v + w, code) then
                isKernel ← false
                break
        if isKernel then
            kernel.append(v)
    return kernel
```

5.6 Shape

En aquesta seccio descriurem el procés per a construir una funció que calcula el *shape* d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ donat.

5.6.1 code_T

Funció que donat un codi $\mathcal{C}$ de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$, determina el conjunt $T(\mathcal{C})$.

Paràmetres d'entrada

- *code*: codi $\mathcal{C}$.
- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- T: conjunt $T(\mathcal{C})$.

Algorisme

Algorithm 10 code_T

```
function CODE_T(code,k1,k2,k3)
    T ← []
    for w ∈ code do
        if w^2 == e_{k1,k2,k3} then
            T.append(w)
    return T
```

5.6.2 code_center

Funció que donat un codi $\mathcal{C}$ en calcula el conjunt centre, $Z(\mathcal{C})$.

Paràmetres d'entrada

- *code*: codi $\mathcal{C}$.

Paràmetre de sortida

- Z: conjunt $Z(\mathcal{C})$.

Algorisme

Algorithm 11 code_center

```
function CODE_CENTER(code)
    Z ← []
    for z ∈ code do
        isCenter ← true
        for x ∈ code do
            if z * x! = x * z then
                isCenter ← false
                break
        if isCenter then
            Z.append(z)
    return T
```

5.6.3 code_type

Funció que retorna el tipus, és a dir la tripleta (σ, δ, ρ), d'un codi determinat.

Paràmetres d'entrada

- *code*: codi del qual en volem calcular el tipus.

Paràmetre de sortida

- $[\sigma, \delta, \rho]$: array que té, en la primera posició, el paràmetre *sigma* de *code*, en la segona, el paràmetre δ i en la tercera el paràmetre ρ.

Algorisme

Algorithm 12 code_type

function CODE_TYPE(*code*,$k1$,$k2$,$k3$)
  $T \leftarrow code\_T(code, k1, k2, k3)$
  $Z \leftarrow code\_center(code)$
  $sigma \leftarrow dim(T)$
  $delta \leftarrow dim(T)/dim(Z)$
  $rho \leftarrow dim(Z)/dim(code)$
  return $[sigma, delta, rho]$

5.6.4 code_shape

Funció que calcula i retorna el *shape* d'un codi $\mathcal{C}$ determinat.

Paràmetres d'entrada

- *code*: codi $\mathcal{C}$.
- *generating_set*: conjunt generador de $\mathcal{C}$.
- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- *shape*: *shape* calculat de $\mathcal{C}$.

Algorisme

(Veure pàgina següent.)

Algorithm 13 code_shape

```
function CODE_SHAPE(code,generatingSet,k1,k2,k3)
    [sigma, delta, rho] ← code_type(code, k1, k2, k3)
    if delta == 0 then
        z1 = generatingSet[rho + delta]
        z2 = generatingSet[rho + delta + 1]
        if z1 == z2 AND z1^2 == (z1, z2) AND z1^2 = u then
            condition ← true
            for i ∈ {0, 1} do
                zi = generatingSet[rho + delta + i]
                for j = 2; j ≤ rho − 1; j++ do
                    zj = generatingSet[rho + delta + k]
                    if (zi, zj)! = zj^2 OR (zi, zj)! = e then
                        condition ← false
                        break
            if condition then
                shape ← 2
        else
            if z1^2 == u then
                condition1 ← true
                for i = 1; i ≤ rho − 1; i++ do
                    zi ← generatingSet(rho + delta + i)
                    if (z1, zi) != zi^2 then
                        condition1 ← false
                        break
                if condition1 then
                    condition2 ← true
                    for i = 1; i ≤ rho − 1; i++ do
                        for j = 1; j ≤ rho − 1; j++ do
                            if i != j AND (zi, zj) != e then
                                condition2 ← false
                                break
                    if condition2 then
                        shape ← 3
            else
                if rho == 4 then
                    shape ← 5
    else
        if rho == 0 then
            shape ← 1
        else
            if rho == 2 AND delta ≤ 1 then
                shape ← 4
            else
                shape ← −1
    return shape
```

5.7 Altres funcions

5.7.1 get_identity

Funció que retorna l'element neutre **e** d'un codi $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.

Paràmetres d'entrada

- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- e: element neutre del codi.

Algorisme

Algorithm 14 get_identity

```
function GET_IDENTITY(k1,k2,k3)
    l ← k1 + k2 + k3
    e ← zeros(1, l)
    for i = k1 + k2; i ≤ l − 1; i++ do
        e[i] = 1
    return e
```

5.7.2 get_second_order

Funció que retorna l'element de segon ordre **u** d'un codi $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$.

Paràmetres d'entrada

- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- u: element de segon ordre del codi.

Algorisme

Algorithm 15 get_second_order

```
function GET_SECOND_ORDER(k1,k2,k3)
    for i = 0; i ≤ k1 − 1; i++ do
        u[i] = 1
    for i = k1; i ≤ k1 + k2 − 1; i++ do
        u[i] = 2
    for i = k1 + k2; i ≤ k1 + k2 + k3 − 1; i++ do
        u[i] = aa
    return u
```

5.7.3 wexp

Funció que calcula w^h.

Paràmetres d'entrada

- w: paraula codi que serveix com a base.
- h: exponent al qual elevem h.
- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- *result*: resultat de l'operació w^h.

Algorisme

Algorithm 16 wexp

function WEXP(w,h,$k1$,$k2$,$k3$)
  for $i = 0;\ i \leq k1 + k2 - 1;\ i++$ **do**
    $result[i] = h * w[i]$
  for $i = k1 + k2;\ i \leq k1 + k2 + k3 - 1;\ i++$ **do**
    $result[i] = w[i]^h$
  return $result$

5.7.4 commutator

Funció que retorna el resultat del commutador de $x, y\ \in \mathcal{C}$ de $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times \mathcal{Q}_8^{k_3}$, (x, y).

Paràmetres d'entrada

- x: primer operand.
- y: segon operand.
- $k1$: número que denota la dimensió k_1 de $\mathbb{Z}_2$.
- $k2$: número que denota la dimensió k_2 de $\mathbb{Z}_4$.
- $k3$: número que denota la dimensió k_3 de $\mathcal{Q}_8$.

Paràmetre de sortida

- Retorna el resultat de la operació commutador.

Algorisme

Algorithm 17 commutator

function COMMUTATOR(x,y,$k1$,$k2$,$k3$)
  return $x^{-1} * y^{-1} * x * y$

5.8 Test

L'avaluació de les funcions creades ha estat pràcticament teòrica. A més, s'han utilitzat d'altres funcions externes, que es detallen en els subapartats següents. D'aquestes funcions no en detallarem l'algorisme, donat que no han estat implementades per nosaltres.

5.8.1 hadamard_matrix

Funció que intenta construir una matriu de Hadamard utilitzant una combinació de construccions de Paley i Sylvester.

Paràmetres d'entrada

- n: dimensió de la matriu que volem generar.

Paràmetre de sortida

- Retorna, si s'ha pogut construir, una matriu de Hadamard de dimensió n. Si s'ha produït algun error, llança una excepció.

6 Resultats

En aquest capítol s'haurien de veure els resultats obtinguts amb el desenvolupament de la llibreria. Com ja s'ha anat explicant en capítols anteriors, aquest projecte pretén establir les bases per a futures investigacions sobre els codis $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.

Per les funcions implementades no té sentit presentar-ne resultats escrits, doncs són funcions fins ara no implementades. Per veure'n els seus resultats hom hauria de descarregar-se el paquet i utilitzar-lo en un entorn de Sage.

7 Conclusions

En aquest projecte es presenta el desenvolupament d'una llibreria d'aplicacions en l'entorn de programació matemàtica Sage per al tractament de codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.

Com ja s'ha vist en seccions anteriors, els codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ són un tipus de codis molt particulars i amb una funció molt teòrica. El treball desenvolupat en aquest projecte senta les bases per a futures investigacions sobre les propietats d'aquests codis.

7.1 Objectius complerts

S'han assolit els objectius estimats al principi del projecte:

- S'han assolit els coneixements necessaris per a la implementació del conjunt de funcions de la llibreria.
- S'ha construït una funció que determina si un codi $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ és Hadamard o no.
- S'ha construït una funció que calcula el *shape* d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.
- S'ha construït una funció que calcula el *rank* d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.
- S'ha construït una funció que calcula el *kernel* d'un codi Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.

7.2 Millores i futures investigacions

Un cop sentades les bases per a la creació dels codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ podem preveure dos camins de millora: l'optimització i la construcció de codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$.

Pel que fa a l'optimització, seria interessant veure si els càlculs implementats en aquest projecte poden rebaixar-se de categoria. Tot i que no s'ha tractat l'optimització com un tema específic d'aquest projecte, sí que s'ha treballat tenint en compte un cert nivell d'optimització a l'hora de programar les funcions de la llibreria. Existeixen, però, alguns aspectes que podrien ésser susceptibles de millora.

A més a més, la llibreria implementada facilita la construcció de codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ i la possibilitat de fer investigacions sobre ells. En l'article [2] es donen indicacions per a realitzar alguna d'aquestes construccions.

7.3 Conclusions finals

Aquest ha estat un projecte complex, sobretot pel que fa al volum d'informació i conceptes teòrics que s'han hagut d'assolir. Aquesta ha estat potser la part més farragosa del projecte tant pel que fa a la interiorització com a l'escriptura. Les bases que s'han assolit en aquesta part però, eren completament necessàries i a més a més, la base matemàtica que portava m'ha fet més fàcil assolir algun dels conceptes més teòrics.

Un cop assolides les bases, el procés d'investigació i desenvolupament ha estat molt interessant. El fet d'haver de buscar la solució i de planificar els temps d'entrega, han donat un plus al projecte que he trobat molt enriquidor.

Espero que en el futur es puguin desenvolupar i investigar nous codis sobre la base dels codis Hadamard $\mathbb{Z}_2\mathbb{Z}_4\mathcal{Q}_8$ i potser, que siguin prou interessants per mirar d'aplicar-los en algun projecte pràctic.

Annex

Apèndix A Diagrama de Gantt

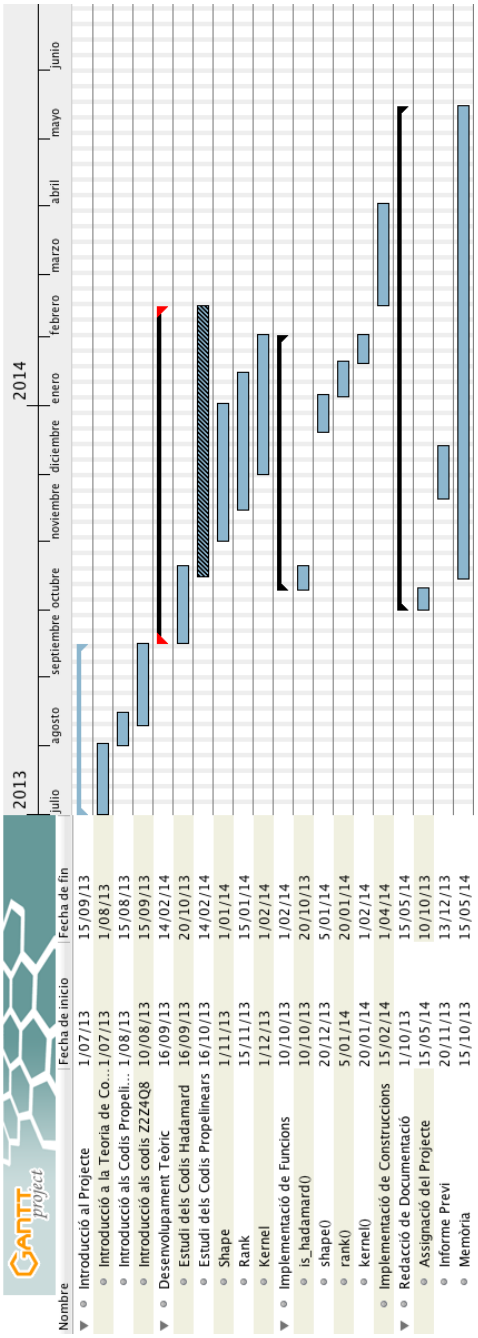

Figura 2: Diagrama de Gantt

8 Referències

[1] J. Borges, C. Fernández, J. Pujol, J. Rifà, and M. Villanueva. $\mathbb{Z}_2\mathbb{Z}_4$-linear codes: generator matrices and duality. *Desigus, codes and cryptography*, 54:167–179, January 2010.

[2] A. del Rio and J. Rifà. Families of hadamard $\mathbb{Z}2\mathbb{Z}4\mathcal{Q}8$-codes. *IEEE Trans. on Information Theory*, 59:5140–5151, August 2013.

[3] A. del Rio and J. Rifà. Hadamard quaternionic code. In *Real Sociedad Matemática Española, Matemáticas de la Teoría de la Información*, January 2013.

[4] A.R. Hammons, P.V. Kumar, A.R. Calderbank, N.J.A. Sloane, and P.Solé. The $\mathbb{Z}_4$-linearity of kerdock, preparata goethals and related codes. *IEEE Trans. on Information Theory*, 40:301–319, 1994.

[5] J. Rifà, J. Basart, and L. Huguet. On completely regular propelinear codes. In *Lecture Notes in Computer Science*, volume 357, pages 341–355, 1989.

[6] J. Rifà and J. Pujol. Translation invariant propelinear codes. *IEEE Trans. on Information Theory*, 43:590–598, March 1997.

[7] William Stein. SAGE Days 4. `http://web.archive.org/web/20070627235122/http://www.sagemath.org/why/stein-sd4.pdf`, 2007.